

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

Publications from USDA-ARS / UNL Faculty

U.S. Department of Agriculture: Agricultural
Research Service, Lincoln, Nebraska

2008

Building a Community Modelling and Information Sharing Culture

A. Voinov

Chesapeake Research Consortium

R. R. Hood

HPL/UMCES

J. D. Daues

USDA-ARS, Jornada Experimental Range

H. Assaf

American University of Beirut

R. Stewart

University of Tennessee

Follow this and additional works at: <https://digitalcommons.unl.edu/usdaarsfacpub>



Part of the [Agricultural Science Commons](#)

Voinov, A.; Hood, R. R.; Daues, J. D.; Assaf, H.; and Stewart, R., "Building a Community Modelling and Information Sharing Culture" (2008). *Publications from USDA-ARS / UNL Faculty*. 410.
<https://digitalcommons.unl.edu/usdaarsfacpub/410>

This Article is brought to you for free and open access by the U.S. Department of Agriculture: Agricultural Research Service, Lincoln, Nebraska at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Publications from USDA-ARS / UNL Faculty by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.



CHAPTER TWENTY

BUILDING A COMMUNITY MODELLING AND INFORMATION SHARING CULTURE

A. Voinov ^a, R.R. Hood ^b, J.D. Daues ^c, H. Assaf ^d, and R. Stewart ^e

Contents

20.1. Introduction	345
20.2. Open Source and Hacker Culture	347
20.3. Knowledge Sharing and Intellectual Property Rights	350
20.4. Software Development and Collaborative Research	352
20.5. Open Source Software vs. Community Modelling	353
20.6. Pros and Cons of Open-Source Modelling	357
20.7. Open Data	360
20.8. Teaching	361
20.9. Conclusions and Recommendations	362
References	365



20.1. INTRODUCTION

Much of human creativity is geared towards moving energy and materials rather than information, even though information has become another crucial component of human welfare and livelihood. Information, unlike energy and materials, is not subject to conservation laws. By copying information from sources and distributing it to new destinations we do not lose information at the sources. This is what is known as non-rival goods in ecological economics (Daly and Farley, 2003). As with gravity, by using information we do not decrease the ability of others to use it. Nevertheless, exchange of information is restricted by patent law, as well as by institutional, cultural and traditional hurdles that create protective barriers hindering the free flow of this valuable commodity. In this way we are making it

^a Chesapeake Research Consortium, 645 Contees Wharf Road, PO Box 28, Edgewater, MD 21037, USA
^b HPL/UMCES, PO Box 775, Cambridge MD 21613, USA
^c USDA-ARS-Jornada Experimental Range, Las Cruces, NM, USA
^d Department of Civil and Environmental Engineering, American University of Beirut, PO Box 11-0236, Riad El Solh, Beirut 1107 2020, Lebanon
^e University of Tennessee, The Institute for Environmental Modeling, 569 Dabney Hall 1416, Circle Drive Knoxville, TN 37996-1610

excludable. It is not surprising that private companies are often reluctant to share data and software because it can impact their profits in a competitive market.

Unfortunately, barriers to information exchange are also significant in the academic community, where the long-standing emphasis on publication and (perhaps unwarranted) fear of misuse of released data and software have inhibited free and open exchange. Promotion and tenure at academic institutions is still largely dependent upon the volume of peer-reviewed publications and success in securing grant and contract funds. As a result, academic scientists have little or no incentive to spend the time and effort that are required to document and disseminate their data and/or their models and code for the greater good of the research community. This problem is exacerbated by the fact that grant and contract funding for research rarely provides direct support for documentation and dissemination activities. The issue is particularly acute when it comes to sharing the source code of models and data analysis software. Even if a scientist or engineer is amenable to sharing the code, the effort required to provide documentation to make it useful is often viewed as an insurmountable obstacle.

Funding agencies worldwide seem to clearly recognise the pressing need to enhance communication and promote open exchange of data and information among scientists and between academic and private institutions via the Internet. The National Science Foundation, for example, has initiated several new major research initiatives that are aimed at developing and/or explicitly requiring this enhanced communication. These initiatives include NEON (National Ecological Observatory Network), CLEANER (Collaborative Large-Scale Engineering Analysis Network for Environmental Research), CUASHI (Consortium of Universities for the Advancement of Hydrological Sciences, Inc.), and ORION (Ocean Research Interactive Observatory Network), to name just a few. The European Union has funded such open-source projects as Harmon-IT and SEAMLESS. All of these initiatives embrace the idea that developing the infrastructure needed to allow free and open exchange of large volumes of data and information will be crucial for making rapid scientific advancements in the future. For example, the success of current efforts to develop earth observatories in both terrestrial (e.g. NEON) and marine (e.g. ORION) environments will be critically dependent upon the successful development of this infrastructure because these observatories will have to collect, process and disseminate large volumes of data and assimilate them into models in a timely manner.

The challenges we face in creating a new research paradigm are many. Substantial improvements in hardware (e.g. network and computing infrastructure), software (e.g. data base manipulation software, and data assimilating numerical models), and a much higher level of standardisation of data formats will be required. New means for carrying out real-time data processing and automated data quality control will also have to be developed. However, we believe that one of the greatest challenges we face in this endeavour will be building the community modelling and information sharing culture that will be required for success. How do we get engineers and scientists to put aside their traditional modes of doing business? How do we provide the incentives that will be required to make these changes happen? How do we get our colleagues to see that the benefits of sharing resources far outweigh the costs?

We argue that timely sharing of data and information is not only in the best interest of the research community, but that it is also in the best interest of the scientist who is doing the sharing. Substantial additional benefits will be derived through new contacts, collaborations and acknowledgment that are fostered by open exchange. Numerous examples attest to this fact, some of which are described below. The real challenge we face is getting our colleagues to recognise the potential benefits that can be derived from adopting a community modelling and information sharing culture. In addition, we need to dispel unwarranted fears that many scientists and engineers harbour, i.e. that they will be “scooped” if they release their data too soon or blamed if there is a bug in their code. And finally, we need to accept the fact that releasing undocumented or poorly documented software is a preferable alternative to not releasing it at all.

In the following pages we discuss the history of the open-source movement, focusing primarily on software development. This movement has its origins in “hacker” culture, and it matured in the software development community as a sophisticated and efficient means for developing software. This culture has now penetrated virtually every aspect of software development and it is certainly applicable to both information and data sharing. Although the scientific community has been slow to adopt it, we believe that building the community modelling and information sharing culture among scientists will be crucial for future advancement in environmental and earth science.



20.2. OPEN SOURCE AND HACKER CULTURE

Computer programming in the 1960s and 1970s was dominated by the free exchange of software (Levy, 1984). This started to change in the 1980s when the Massachusetts Institute of Technology (MIT) licensed some of the code created by its employees to a commercial firm and also when software companies began to impose copyrights, and later software patents, to protect their software from being copied (Drahos and Braithwaite, 2002).

Probably in protest to these developments, the open-source concept started to gain ground in the 1980s. The open-source concept stems from the so-called hacker culture. Hackers are not what we usually think they are – software pirates, vicious producers of viruses, worms and other nuisances for our computers. Hackers will insist that those people should be called “crackers.” Hackers are the real computer gurus, who are addicted to problem solving and building things. They believe in freedom and voluntary mutual help. It is almost a moral duty for them to share information, solve problems and then give the solutions away just so other hackers can solve new problems instead of having to re-address old ones. Boredom and drudgery are not just unpleasant but actually evil. Hackers have an instinctive hostility to censorship, secrecy, and the use of force or deception.

The idea of software source code shared for free is probably best known in connection with the Linux operating system. After Linus Torvalds developed its core and released it to software developers worldwide, Linux became a product of joint

efforts of many people, who contributed code, bug reports, fixes, enhancements and plug-ins. The idea gained momentum when Netscape released the source code of its Navigator, the popular Internet browser program in 1998. That is when the term “open source” was coined and when the open-source definition was derived. Both Linux and Navigator (the latter now developed as the Firefox browser under mozilla.org) have since developed into major software products with worldwide distributions, applications and input from software developers (Bollier, 1999).

“The basic idea behind open source is very simple: When a programmer can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing” (Raymond, 2000a). Motivated by the spirit of traditional scientific collaboration, Richard Stallman, then a programmer at MIT’s Artificial Intelligence Laboratory, founded the Free Software Foundation (FSF) in 1985 (<http://www.fsf.org/>). The FSF is dedicated to promoting computer users’ rights to use, study, copy, modify and redistribute computer programs. Bruce Perens and Eric Raymond created the Open Source Definition in 1998 (Perens, 1998). The General Public License (GPL), Richard Stallman’s innovation, is sometimes known as “copyleft.” A form of copyright protection achieved through contract law. As Stallman describes it: “To copyleft a program, first we copyright it; then we add distribution terms, which are a legal instrument that gives everyone the rights to use, modify, and redistribute the program’s code or any program derived from it, but only if the distribution terms are unchanged.” The GPL creates a commons in software development “to which anyone may add, but from which no one may subtract.”

One of the crucial parts of the open-source licence is that it allows modifications and derivative works, but all of them must be then distributed under the same terms as the license of the original software. Therefore, unlike simply free code that could be borrowed and then used in copyrighted, commercial distributions, the open-source definition and licensing effectively makes sure that the derivatives stay in the open-source domain, extending and enhancing it. The GPL prevents enclosure of the free software commons and creates a legally protected space for it to flourish. Because no one can seize the surplus value created within the commons, software developers are willing to contribute their time and energy to improving it. The commons is protected and stays protected.

The GPL is the chief reason that Linux and dozens of other programs have been able to flourish without being privatised. The Open Source Software (OSS) paradigm can produce innovative, high-quality software that meets the needs of research scientists with respect to performance, scalability, security, and total cost of ownership (TCO). OSS dominates the Internet with software such as Sendmail, BIND (DNS), PHP, OpenSSL, TCP/IP, and HTTP/HTML. Many excellent applications also exist including Apache web server, Mozilla Firefox web browser and Thunderbird email client, the OpenOffice suite, and many others (Wheeler, 2005).

OSS users have fundamental control and flexibility advantages. For example, if one were to write a model using ANSI standard C++ (as opposed Microsoft C++), one could easily move the code from one platform to another. This may be convenient for a number of reasons, from simply a preference from one developer to

another, to moving from a desktop PC environment to a high performance computing environment. Open Standards, which are publicly available specifications, offer control and flexibility as well. Examples in science include Environmental Markup Language (EML) and Virtual Reality Markup Language (VRML). If these were proprietary, use would be likely limited to one propriety application to interface with one proprietary format or numerous applications, each with its own format. One need only imagine the limitations on innovation if commonly used protocols like ASCII, HTTP, or HTML were proprietary. To organise this growing community the Open Source Development Network (OSDN) (<http://www.osdn.com>) was created. Like many previous open-source spin-offs, it is based on the Internet and provides the teams of software developers distributed around the world with a virtual workspace, where they can discuss their ideas, progress, bugs, share updates and new releases. The open-source paradigm has become the only viable alternative to the copyrighted, closed and restricted corporate software.

What underlies the OSS approach is the so-called “Gift culture” and “Gift economy” that is based on this culture. Under Gift Culture you gain status and reputation in it, not by dominating other people, nor by being special or by possessing things other people want, but rather by giving things away. Specifically, by giving away your time, your creativity, and the results of your skill. We can find this in some of the primitive hunter–gatherer societies where a hunter’s status was not determined by how much of the kill he ate, but by what he brought back for others. One example of a gift economy is the potlatch, which is part of the pre-European cultures of the Pacific Northwest of North America. In the potlatch ceremony, the host demonstrates his wealth and prominence by giving away possessions, which prompts participants to reciprocate when they hold their own potlatch. There are many other examples of this phenomenon. What is characteristic of most is that they are based on abundance economies. There is usually a surplus of something that is easier to share than to keep for yourself. There is also the understanding of reciprocity that by doing this people can lower their individual risks and increase their survival (Raymond, 2000a).

In hunter–gatherer societies, freshly killed game called for a gift economy because it was perishable and there was too much for any one person to eat. Information also loses value over time and has the capacity to satisfy more than one. In many cases information gains rather than loses value through sharing. Unlike material or energy, there are no conservation laws for information. On the contrary, when divided and shared, the value of information only grows. The teacher does not know less when he shares his knowledge with his students. While the exchange economy may have been appropriate for the industrial age, the gift economy is coming back as we enter the information age.

It should be noted that the community of scientists, in a way, follows the rules of a gift economy. The scientists with highest status are not those who possess the most knowledge; they are the ones who have contributed the most to their fields. A scientist of great knowledge, but only minor contributions is almost pitied – his or her career is seen as a waste of talent. But in science the gift culture has not yet fully penetrated to the level of data and source code sharing (Hippel and Krogh, 2003). This culture has been inhibited by an antiquated academic model for promotion

and tenure that is still prevalent today. This culture encourages delaying release of data and source code to ensure that credit and recognition are bestowed upon the scientist who collected the data and/or developed the code. This model (which was developed when data were much more difficult to collect and analyse and long before computers and programming existed) no longer applies in the modern scientific world where new sensor technologies and observing systems generate massive volumes of data and where computer programs and numerical models have become so complex that they cannot be fully analysed or comprehended by one scientist or even by small teams.



20.3. KNOWLEDGE SHARING AND INTELLECTUAL PROPERTY RIGHTS

The concept of intellectual property rights and the enactment of laws to protect them were first formalised in the Statute of Anne that was passed by the British Parliament in the early 18th century in an attempt to stem the rapid rise in unauthorised printing of books facilitated by the advent of affordable and efficient printing technology (Tuomi, 2004). Formally, an intellectual property (IP) is a knowledge product that could be an idea, a concept, a method, an insight or a fact that is manifested explicitly in a patent, copyrighted material or some other form, where ownership can be defined, documented, and assigned to an individual or corporate entity (Howard, 2005).

Although the concept of public domain was implicitly considered by the Statute of Anne, it was clearly articulated by Denis Diderot who was retained by the Paris Book Guild to draft a treatise on literary rights. In his “Encyclopedie,” Diderot advocated the systemic presentation and publication of knowledge of all the mechanical arts and manufacturing secrets for the purpose of reaching the public at large, promotion of research and weakening the grip of craft guild on knowledge (Tuomi, 2004). With these pioneering ideas, Diderot set the stage for the involvement of public domain, which includes non-exclusive IP that is freely, openly available and accessible to any member of the society.

Public domain and exclusive IP rights represent the two extremes in IP regimes, with the former providing a free sharing of knowledge and the latter emphasising the rights of owners in limiting access to their knowledge products. Since the inception of the concept of intellectual property rights, it was argued that protecting these rights provides adequate compensations for owners and encourages innovations and technological development. However, historical evidence and published research does not support this claim and points to lack of concrete evidence that confirms these claims (National Academy of Engineering, 2003). Also increasingly many technological innovations are the result of collaborative efforts in an environment that promotes non-exclusive intellectual rights. Although most of these efforts are in the software development domain, e.g. development of Linux, it is interesting to note that the tremendous growth and development in the semi-conductor industry is mainly attributed to the highly dynamic and connected social networks

of Silicon Valley in the 1960s, which was regarded as a public domain region, since information and know-how were freely shared among its members.

In the world of business, preservation of exclusive IP rights is seen as a necessity to maintain competitive edge and protect expensively obtained technology. Patents that were designed to stimulate innovation are now having the opposite effect, especially in the software industry. As Perens describes: "Plagued by an exponential growth in software patents, many of which are not valid, software vendors and developers must navigate a potential minefield to avoid patent infringement and future lawsuits" (Perens, 2006a). The big corporations seem to solve the problem by operating in a "detente" mode: by accumulating huge numbers of patents themselves they become invulnerable to claims from rivals, i.e. competitors do not sue out of fear of reciprocity. However, now we see that whole companies are created with the sole purpose of generating profit from patents. These "patent parasites" make no products and derive all of their income from patent litigation. Since they make no products, the parasites themselves are invulnerable to patent infringement lawsuits, and can attack even very large companies without any fear that those companies will retaliate. One of the most extreme and ugly methods is known as patent farming: influencing a standards organisation to use a particular principle covered by a patent. In the worst and most deceptive form of patent farming, the patent holder encourages the standards organisation to make use of a principle without revealing the existence of a patent covering that principle. Then, later on, the patent holder demands royalties from all implementers of the standard (Perens, 2006b).

Certainly these patent games are detrimental for small businesses. According to the American Intellectual Property Law Association, software patent lawsuits come with a defence cost of about \$3 million per annum. A single patent suit could bankrupt a typical small or medium-size applications developer (let alone an open-source developer) even before the case is fully heard (Newscom, 2005). The smaller patent holder simply cannot sustain the expense of defence, even when justified, and is forced to settle and license patents to the larger company. The open-source community is also constantly under the threat of major attacks from large corporations. There is good reason to expect that Microsoft will soon be launching a patent-based legal offensive against Linux and other free software projects (Newsforge, 2004).

Unfortunately, universities are increasingly seeking to capitalise on knowledge in the form of IP rights. However, only a few of these universities are generating significant revenues from licensing IP rights (Howard, 2005). This equally applies to individual researchers who may seek to protect and profit from their findings. Interestingly, Howard (2005) reports that research conducted by the Association for Institutional Research in the United States (Owen-Smith and Powell, 2000) shows a marked difference in how researchers from different disciplines perceive IP rights and the prospect of patenting. Physical scientists from natural and engineering schools expect less personal gain from patent royalties, favour non-exclusive licence arrangements where they rely more on providing service or consultancy, and are less concerned about identifying the proper IP license. On contrast, life scientists expect more personal gain from patent royalties, favour exclusive licensing arrangements and are more concerned about protecting IP. One possible explanation for this is that over time there have been so many more patents issued in the physical and

engineering domains that a certain saturation level might be approaching, while patenting is still relatively new to the life sciences.



20.4. SOFTWARE DEVELOPMENT AND COLLABORATIVE RESEARCH

Just as public domain and exclusive IP rights represent the two extremes in IP regimes, the software development process can occur in one of two ways, either the “cathedral” or the “bazaar” (Raymond, 2000a). The approach of most producers of commercial, proprietary software is that of the cathedral, carefully crafted by a small number of people working in isolation. This is the traditional approach we also find in scientific research. Diametrically opposed to this is the bazaar, the approach taken by open-source projects. Open source encourages people to freely tinker with the code, thus permitting new ideas to be easily introduced and exchanged. As the best of those new ideas gains acceptance, it essentially establishes a cycle of building upon and improving the work of the original coders, frequently in ways they did not anticipate. The release process can be described as release early and often, delegate everything you can, be open. Leadership is essential in the OSS world, i.e. most projects have a lead that has the final word on what goes in and what does not. For example, Linus Torvalds has the final say on what is included in the kernel of Linux. In the cathedral-builder view of programming, bugs and development problems are tricky, insidious, deep phenomena. It takes months to weed them all out. Thus the long release intervals, and the disappointment when long-awaited releases are not perfect. In the bazaar view, most bugs turn shallow when exposed to a thousand co-developers. Accordingly you release often in order to get more corrections, and as a beneficial side effect you have less to lose if a bug gets out the door.

It is clear that the bazaar approach can work in general scientific projects and in modelling applications in particular. Numerous successful examples, especially in earth system modelling, attest to this fact. But we must also recognise that there is a difference between software development and science, and that software engineers and scientists have different attitudes about software development. For a software engineer, the exponential growth of computer performance offers unlimited resources for the development of new modelling systems. Engineers therefore view models as just pieces of software that can be therefore built from blocks or objects, almost automatically and then connected over the web and distributed over a network of computers. It is simply a matter of choosing the right architecture and writing the appropriate code. The code is either correct or not, either it works or crashes. Not so with a scientific model. Rather, most scientists consider that a model is useful only as an eloquent simplification of reality that needs profound understanding of the system to be built. A model should tell us more about the system, than simply the data available. Even the best model can be wrong and yet quite useful if it enhances our understanding of the system. Moreover, it often takes a long time to develop and test a scientific model.

As a result of this difference in point of view and approach, we tend to see much more rapid development of new languages, software development tools and open code and information sharing approaches among software engineers. In contrast, we see relatively slow adoption of these tools and approaches by the research modelling community. This is in spite of the fact that they will undoubtedly catalyse more rapid scientific advancements. As web services empower researchers, it is becoming clear that the biggest obstacle to fulfilling this vision of free and open exchange among scientists is cultural. Competitiveness and conservative approaches will always be with us, but developing ways to give meaningful credit to those who share their data and their code will be essential in order to change attitudes and encourage the diversity of means by which researchers can contribute to the global academy (Nature, 2005). It is clear that a new academic model that promotes open exchange of data, software and information is needed. Fortunately, the success of the open-source approach in software development has instigated researchers to start considering similar, shared, open approaches in scientific research. Numerous collaborative research projects are now based on Internet communications and are led simultaneously at several institutions working on parts of a larger endeavour (Schweik et al., 2005). Sometimes such projects are open and allow new researchers to participate in the work. Results and credit are usually shared among all the participants. This trend is being fuelled by the general trend of increasing funding for large collaborative research projects, particularly in the earth sciences.



20.5. OPEN SOURCE SOFTWARE VS. COMMUNITY MODELLING

The recent emergence of open-source model development approaches in a variety of different earth science modelling efforts (which we refer to here as community modelling) is an encouraging development. Although the basic approach is the same, we can also identify several aspects of research-oriented community modelling that distinguish it from open-source software development. For example, there has been a number of successful community modelling efforts (Table 20.1). However, unlike most open-source software development projects, these have been blessed by substantial grant and contract support (usually from federal sources), and exist largely as umbrella projects for existing on-going research. It is probably also fair to say that most of the existing earth science community models are not truly “open source,” i.e. access to the codes and rules governing modification and redistribution are usually more restrictive than, for example, those under GPL.

In general, in community modelling there is usually a much smaller number of participants because the research community is much smaller and more specialised than the broad field of software development. Because the pool is smaller it may be harder to find the right people, both in terms of their skills and their willingness to collaborate within an open modelling paradigm. Similarly, there is generally a much smaller number of users of open-source research-oriented models, which may be very specialised and usually require specific skills to use. This is mostly because scientific models are very often focused on simulating a specific phenomena

Table 20.1 Some of the on-going community modelling projects

Name	Web site and players	Scope	Projects
CMAS Community Modelling and Analysis System	http://www.cmascenter.org/ Funding – US EPA, Lead – Carolina Environmental Program at the University of North Carolina at Chapel Hill	Development of Air Quality and Meteorological models, extensions of the Models-3/CMAQ. Outreach, user-support	CMAS-Supported Products: Community Multiscale Air Quality (CMAQ) Modelling System, Meteorology Chemistry Interface Processor (MCIP), Sparse Matrix Operator Kernel Emissions (SMOKE), System Package for Analysis and Visualisation for Environmental (PAVE), data Input/Output Applications Programming Interface (I/O API), MM5 Meteorology Coupler (MCPL), Multimedia Integrated Modelling System (MIMS)
ESMF Earth System Modelling Framework	http://www.esmf.ucar.edu/	High-performance, flexible software infrastructure for use in climate, numerical weather prediction, data assimilation, and other	Earth science applications
CCSM Community Climate System Model	http://www.ccsn.ucar.edu/ – NCAR	Global atmosphere model for use by the wider climate research community	Working Groups: Atmosphere Model, Land Model, Ocean Model, Polar Climate, Biogeochemistry, Paleoclimate, Climate Variability, Climate Change, Software Engineering

(continued on next page)

Table 20.1 (continued)

Name	Web site and players	Scope	Projects
CSTM National Community Sediment-Transport Model	http://woodshole.er.usgs.gov/project-pages/sediment-transport/ – Woods Hole	Deterministic models of sediment transport in coastal seas, estuaries, and rivers	CTSM modules implemented in ROMS and FVCOM hydrodynamic models. Regional applications: Massachusetts Bay, Hudson River, Adriatic Sea
CCMP Chesapeake Community Model Program	http://ccmp.chesapeake.org – Chesapeake Research Consortium	Estuary, river and watershed modelling for water quality in the Chesapeake Bay	Baywide Hydrodynamic models: Quoddy, ROMS, POM, Biogeochemical models: CH3D_biowp, Larvae tracking IBM: CBOLT, Watershed: CBP-HSPF and V5 data
WATER and Environmental Research Systems (WATERS) Network	http://www.cuahsi.org/ http://cleaner.ncsa.uiuc.edu/home/	Hydrologic sciences, complex, large-scale environmental systems, education, outreach, and technology transfer	CUAHSI Consortium of Universities for the Advancement of Hydrologic Science, CLEANER Collaborative Large-Scale Engineering Analysis Network for Environmental Research
CSDMS – Community Surface Dynamics Modeling System	http://csdms.colorado.edu – University of Colorado, Boulder	Integrated software modules for erosion, transport, and deposition of sediment and solutes in landscapes and their repository sedimentary basins	Terrestrial, coastal, marine worknig groups. Groups on Education and Knowledge Transfer, and Cyberinformatics and Numerics

or addressing a specific scientific question or hypothesis, and also because the scientific community is very small compared to the public at large. Along these same lines, research-oriented models are generally more sophisticated and difficult to use than software products that are developed for the public. It is certainly much harder to run a meaningful scenario with a hydrodynamic simulation model, than to aim your virtual gun at a virtual victim and press the “shoot” button in a computer game (though one might argue that to a large extent this difference in difficulty of use has more to do with the primitive state of the user interface of most scientific codes). It is also generally true that scientific codes require more sophisticated documentation and steeper learning curves to master. Documenting scientific models is a real problem, i.e. it is not what researchers normally enjoy doing and the need for doing it is rarely appreciated and funded. On the other hand, documentation is a crucial part of the process if we anticipate others will use and take part in the development of our models.

Open research modelling is also much more than open programming. As we mentioned above, software development has a clear goal, an outcome. The product specifications can be well established and designed. In contrast, research modelling is iterative and interactive. The goal oftentimes gets modified while the project evolves. It is much more a process than a product. It is usually harder to agree on the desired outcomes and the features of the product. In some respects modelling is more like an art than a science. Following this analogy, how do you get several artists together to paint one picture? This is particularly true in ecological modelling where there is no overarching theory to guide model structure and where a variety of different formulations can be used to represent a particular process. These aspects of scientific modelling actually make it highly amenable to open programming approaches, which naturally allow a high degree of flexibility.

A significant impediment to developing open research models is the lack of infrastructure, i.e. there are still few good software tools to support community research and modelling projects. Once again there is an obvious gap between software and application. There is software that potentially offers some exciting approaches and new paradigms to support modularity, data sharing, web access, or flexible organisation – all the major components required for successful model integration and development. The most recent trends in software design are compared to the Lego constructor over the web (Markoff, 2006), exactly what we need for modular models. However, this is yet to be developed and applied to the modelling process, and embedded into the modelling lexicon and practice. Yet another difference is that most research modelling projects take years to develop. This is in contrast to some of the software hacks that can be invented and implemented in a matter of hours, quickly gaining recognition and respect in the software development community. Research is a much slower and tedious process, where small incremental ideas and successes may be very important, but are much harder to document, disseminate and appreciate.

Finally, returning to the central problem, we really need to change the traditional culture and attitudes of research scientists, i.e. promote a shift in the mindset and psychology that drives scientific research. Historically, most science has been driven by individual efforts and talent. Talent and ingenuity of individuals will always be

critical in scientific exploration, but with the growing amount of data, knowledge and information, most of the breakthrough achievements are now produced in team efforts, where teams and teamwork rather than individuals are key. This trend is being driven to a large extent by the increasing emphasis in scientific research on large projects aimed at solving complex interdisciplinary problems, like simulating and predicting the earth system response to global warming. It is becoming increasingly difficult to identify the sole individual who cried “Eureka!” and solved the problem. Even when it is done, very often the recognition is biased by past success, hierarchy and personalities. There is an obvious need for new award and credit systems that will stimulate sharing and teamwork rather than direct personal gain, credit and fame.



20.6. PROS AND CONS OF OPEN-SOURCE MODELLING

A number of specific arguments are often levelled against the open source modelling approach. Some are deeply rooted in concerns for economic and professional viability and are direct outcomes of established funding approaches and pseudo-competitive business models even within an academic community. Others are based in practical concerns about code reusability, accuracy and applicability. We present here a set of practical concerns in using open source and suggest some approaches for addressing them in a question and answer format.

Q. Pseudo-competitive environment. We have invested considerable effort into this product and it is now on the cutting edge of the field. Why should we openly release the source code? Among sectors of the scientific community there is a competitive or pseudo-competitive environment where we must compete for grant and contract dollars to advance our research and provide support for faculty and staff. Under these circumstances assets, like our source code, can take on real value and provide a competitive edge over other institutions and research groups. Where is the incentive to openly release our code? It may be a very real risk to original creators to release their code as open source as other, perhaps better positioned groups, may then secure future funding. With equal accessibility, expertise and original authorship may be trumped in a future opportunity by outside factors such as differences in cost rates and political landscapes.

A. This is an entirely voluntary decision that works for some, and does not for others. The main point is that open-source research can flourish only when there is no competition for the exact product and result. The code is given to the society at large, not just the competitor. You may gain more by getting so many more people involved in your project and coming up with new solutions, than lose by sharing your secrets. Especially since these secrets are unlikely to last for long. In a way you are claiming ownership of the whole area, idea, domain, and gaining recognition for doing that. This may prove much more beneficial in the long run than holding ownership of a few tricks or solutions for a short period of time. Whatever new code others produce based on yours, it will have your signature.

The increase in professional prestige may easily offset any disadvantage incurred by an open release. In some cases, the software is merely a tool and the group's expertise is the economic driver taking the form of training and consulting services. In this scenario, open source may prove a positive outcome as it extends the prestige of the group and permits rapid development of their existing code by others. This in turn provides new demand for training and consulting. For example, HydroLogics that developed the OASIS river modelling system rarely sells just the software package. It is primarily their river planning services that they profit from. It would make perfect sense for them to claim "ownership" of the whole domain of river modelling and optimisation by going open source.

Indeed economic implications of releasing source code are difficult to predict and may vary on a case-by-case basis. Without a clear advantage there may be a strong reluctance to change the status quo and continue to move forward in a proprietary fashion. Unfortunately this guarded approach may tend to emphasise short term viability at the expense of long term gains realised by creating a centre of expertise or clearing house for models and data management systems, freely open and rapidly developed by the larger community. From an economic standpoint, a collaborative effort that involves serious stakeholders may prove more successful in garnering funding by creating a critical mass of expertise that can respond quickly to needs of sponsors and users alike. This is the strength of the open-source approach. A well-organised, perhaps hierarchal open-source community, may provide greater long-term stability, better products, and rapid development that quickly advance the computational strength of the discipline.

Q. Verification or certification. Software that is deployed for a large number of users or an application that serves a critical role in a non-research capacity must be verified with extreme rigour. How reliable is source code that comes from sources with widely varying coding skills and commitment to verification? It is difficult to control even a small group of dedicated professional programmers, in terms of requirement expectations, version control, verification adequacy, code reuse, etc. How does code that has been developed by a largely unknown group of investigators become verified? How can this approach be applied to a focused software project that has a finite time frame, finite budget, and contractual penalties for failing to deliver? Are the original authors of the software expected to provide all the quality control and the stamp of approval? If so, how can this be funded in a traditional academic environment?

A. If we look at the open-source software example, it is the original author or group of authors who usually keep the authority of "releasing" the new version of the product. But usually there are also several versions (alpha, beta) that are in the works. Those are the versions that have additions and changes that have not been tested yet. The power of the paradigm is that help is available not only at the stage of writing new code (or new model components), but also for testing and documenting them. By making the additions available to the whole community we immediately expose it to a much more vigorous testing than any one administrator can provide (Kipp, 2005). No addition becomes part of the next official release until it is tested to the extent needed. Again it is up to the particular community to decide what level of validation is acceptable. This kind of community testing actually works

best for GUIs that have many different options and paths. The more people that get involved in choosing those paths, the more different options get explored. In the case of modelling, better modular architectures that make the model less dependent upon changes that are made in particular pieces or branches are needed.

On the other hand, the open-source approach may not work in all cases. It may not be the best solution for a project on a very restricted time frame and budget, with contractual penalties. However, this does not mean that the product of such a project cannot enter into the open source domain and benefit other developers in the future. At the same time it may eventually be improved and may help developers produce updates and new releases. For example, the HEC suite of models developed by the Army Corps of Engineers is free and open for download. However the models are not open source, exactly out of fear that additions will not be validated and certified to the standards. As a result the limited staff at HEC do not have the capability to address all the concerns and requests that come from the user community, making the models marginally useful in many cases and pushing the users towards the “reinventing the wheel” approach, when entirely new models are built instead of improving the already existing ones.

Q. Release issues. The use of open-source code may require the unplanned release of the developed code in the process. Release, rather, should be planned with long term strategies in place for monitoring community development, approving and integrating new modules, and maintaining the reputation and credibility for the body of code. Typical funding cycles do not include this kind of opportunity and stakeholders may not be interested in a long-term patronage of this kind.

A. The decision to use open source should be made as early as possible in the exploratory or planning stage. Existing projects may need to take pause and evaluate the bottom line implications of switching to open source midstream. A well-informed decision should include a comprehensive survey of existing codes and their licensing requirements. In some cases, licensing may not require the host code to release its source. This type of a priori information can lead to faster and potentially better development through the use of pre-existing, often mature codes. This may create a lighter, less expensive route to the same end. Furthermore, funds that might have been originally planned for code development can now be used for a more comprehensive verification of the final system. This type of approach can result in stable codes and shorter timetables, both of which are important and tangible outcomes of the open source model.

Q. True code reuse. An existing model rarely meets the needs of a particular individual or application. So when starting to use an existing open source model one often must delve deeply into the code. Despite coding standards and conventions, each programmer has their own style that must be understood. Documentation may not exist, or it may be poor. What can be done to facilitate use of open source code in this respect?

A. These are very valid concerns, even more so with models that may be very case-specific and may be hard to understand. We clearly need more work to be done on documentation, protocols, ontologies, data and model formats. However this is not a question for open vs. closed modelling. It is a broader issue about modelling in general, as discussed above. It is a particularly vexing problem for the academic

community where funding for documentation-related activities is rarely available. The major funding agencies are moving toward demanding timely release of data and other products derived from federally-funded projects. We can be hopeful that these agencies will also eventually demand (and fund) documentation-related activities as well.

Q. Stakeholder Concerns. Some of our sponsors do not want to release particular sections of the source code we have developed for them, for example, with regulatory code that has implications regarding public health. There is concern that stakeholders with proper or improper financial incentives may modify this source, not necessarily to cheat but to encourage a preferred outcome. Another example involves codes that have been developed for military applications.

A. OSS is not the best approach for the development of all software. The examples you mentioned here are cases where OSS may not work. However, as noted above, many sponsors like the product to be open source. It can be argued that when the funding comes from tax dollars then the product should be openly available, i.e. in the public domain. Stakeholder manipulation can be easy to track by simply running the officially-released version with the same data set. On the other hand, if code addition is sensitive and should remain secret this does not seem to contradict the GPL, since nobody will see this addition, and the results produced with it. Nobody can prevent you from writing your own additions and keeping them to yourself. The GPL controls only the ways you release them.



20.7. OPEN DATA

In addition to the trend toward open-source modelling in science, there has also been an increasing emphasis on timely data sharing and archiving to prevent loss of valuable information. To a large extent this trend is being driven by new requirements that are being put in place by government research sponsors. For example, the U.S. National Science Foundation (NSF) now requires specific data management plans and time lines for archiving data in permanent repositories such as the NOAA National Oceanographic Data Center (NODC). Once these data are archived, they are available to anyone who wishes to use them. In addition, the trend of increased data sharing is also being driven by the rapidly increasing volumes of data that are generated by increasingly sophisticated and automated observing systems. These include, for example, satellite probes and ground-based continuous monitoring sensors and sensor networks. Thus, our ability to collect and store large volumes of data is pushing science toward an “abundance economy” where there is a surplus of data that cannot possibly be fully analysed and understood by a single individual or small group of scientists. Open data sharing allows scientists to “hack” at information, extracting additional results, applying it to answer new questions and using it in other research programs that may extend far beyond the original goal of the program that generated the data.

For the open data model to provide the maximum value, all applications have to be able to use it, i.e. implementations of the open data model should be plat-

form and application independent. For example, XML makes it possible for the same information to interact with multiple programs in multiple environments. Instead of the information being bound inseparably to one program, it can be read, processed, and stored by any number of programs. The Open Document Format (ODF) is an open document XML file format for saving and exchanging editable office documents (http://en.wikipedia.org/wiki/Document_file_format). The need for easy data sharing is also driving a trend toward increased standardisation of not only data formats, but also data descriptions, i.e. the so-called metadata that allows a researcher to figure out where the data came from, how it was collected and how it is organised. Several organisations (e.g. the Open Data Foundation (<http://www.opendata.us>), the Open Data Format Initiative (<http://odfi.org/>), and the Open Data Consortium) have emerged in the last decade that are dedicated to guaranteeing the free access of citizens to public information, and making sure that the encoding of data is not tied to a single provider. The use of standard and open formats, such as netCDF and HDF, guarantees this free access, and also often necessitates the creation of compatible free software.

The issue of open data becomes especially important because modern governments generate a vast number of digital files every day, from birth certificates and tax returns to criminal DNA records. All of these documents must be retrievable in perpetuity and shared by numerous agencies and departments. As a result, governments have been reluctant to store official records in the proprietary formats of commercial-software vendors and so have already adopted an open data model by necessity. Scientists have been slow to adopt these kinds of standards for a variety of reasons, not the least of which is the understandable desire to retain privileged access to data that they have invested heavily in collecting, pending publication. But times are changing. As we discussed above, there are huge amounts of data that do not need to be kept behind walls. Moreover, it is now possible to make data available under a Creative Commons licence (see <http://creativecommons.org/license>), where both rights and credits for the reuse of data can be stipulated, while allowing its uninterrupted access by machines (Nature, 2005). Unfortunately, very few scientists and academic organisations seem to be aware of this option.



20.8. TEACHING

It makes perfect sense to also consider how the open-source paradigm can be used to advance education (Voinov, 2002). A web-based course can serve as a core for the joint efforts of many researchers, software developers, educators and students. Imagine a web-based course where researchers describe their findings that are appropriate for the course theme and provide open access to their data and models. Educators could organise the course modules into subsets and sequences that best match the requirements of a particular program and curricula. Software developers might contribute open source software tools for visualisation, interpretation and communication. Students would be there to test the materials offered

and to contribute their feedback and questions, which is essential for improvements of both the content and the form of representation.

Much can be learned from textbooks and recorded sources by the students themselves. However, a good teacher is always essential to facilitate and expedite the learning process. Borrowing from the open-source experience of material development, we can also envision a community of educators who participate in teaching a web-based course, logging into the virtual classroom to contribute to the discussions with students, to answer their questions, to grade their exercises. In this case the talents of the best teachers can be made available to the widest possible audience of students. With a sufficient number of qualified volunteers involved, this kind of education can become a free alternative to the increasingly expensive university education. In compliance with the open-source definition the students educated for free would be expected to contribute in the future to this kind of free virtual education, further enhancing the community of educators.

One can easily envision an Open Network for Education (ONE) set up much like the OSDN to promote and organise free open-source education (along with open distribution of related tools and resources) in a variety of disciplines. MIT has already embarked in this direction, announcing that during this decade it will make nearly all course materials available free on the World Wide Web (<http://web.mit.edu/newsoffice/2001/ocw.html>). This new program, known as MIT OpenCourseWare (<http://ocw.mit.edu/index.html>), is a publication of MIT course materials and does not require any registration or fees. So far it is not degree granting and does not involve direct faculty interaction. Once the gift economy concept spreads to education, we can easily imagine an open-source approach based on these or other collections of tutorials, books and lectures.



20.9. CONCLUSIONS AND RECOMMENDATIONS

So how do we do it? How can we apply and extend the highly successful model of open-source software development to open-research modelling, data sharing and education? What is the “scientific” version of hacker’s culture? How can we make something useful beyond our small community (our gift economy)? How do we build a cathedral in the middle of the bazaar?

One of the major challenges we face in this endeavour is overcoming the pervasive reluctance among scientists about releasing data, models and code for fear of getting “scooped.” This reluctance stems from the persistence of traditional modes of carrying out scientific research, i.e. science used to be driven primarily by single-investigator research, when it was much more experimental, and data were much harder to collect. Under those conditions, there is potentially great risk associated with giving away data or a model before full credit has been garnered through publication. This problem is exacerbated by the fact that pursuit of “fame” is a major driver for many scientists, i.e. if you give away your data and your models too quickly then somebody else might publish them first and you will make them famous instead of yourself. Moreover, many scientists do not want to share their

models and code out of fear of others finding their bugs and mistakes. It is safer to keep your code and your data to yourself. Raymond (2000b) points out another difference between the prevailing academic tradition and the hacker culture. In academia, publicly criticising the work of others is an important mode of gaining reputation. In contrast, in hacker culture, such behaviour is heavily tabooed. This culture of criticism is something scientists will perhaps need to change if collaborative research is to gain ground.

But the times are changing. Many of the old rules and fears are not valid anymore in modern scientific research where we are awash in data, where collaborative, multi-investigator teams are the norm rather than the exception, and where models are becoming increasingly complex to address increasingly complex problems. In the modern world of scientific research it clearly makes sense to share data, code and ultimately credit (Bollier, 1999). Unfortunately, universities tend to perpetuate old-fashioned behaviours because most still use traditional criteria for promotion and tenure, i.e. emphasising first author publications, and success in obtaining grants and contracts. There is little top-down incentive to share. Fortunately, the funding agencies are starting to apply pressure to share data in a timely manner, and pressure to share code is likely to soon follow.

Another big part of the problem is that there is a gap between the average scientist using a model that might be written in FORTRAN, for example, and more modern programming languages and approaches. More widespread adoption of open modelling languages that can be easily plugged into (and saved from) open model building frameworks would greatly facilitate open source modelling in research. It would allow scientists to take full advantage of modern open-source software development tools like CVS (Concurrent Versions System, <http://www.nongnu.org/cvs/> – also an open-source project), Subversion, etc. For open-source modelling to become a reality in scientific research, we will need to be able to use the same or similar tools. Fortunately, movement in this direction is being facilitated by the growing need to develop modelling platforms that accept data from the web and that therefore use common standards and formats for geospatial data. Adoption of modern, open-source programming and code-sharing approaches and tools will ultimately make it possible to construct deeper and more complex models and solve deeper and more complex problems.

In addition to the need for developing new methods and approaches that facilitate open development and sharing of models and large volumes of data, there is also a demand for new “process methods” for working with people, communities, and businesses in scientific pursuits. The development of the Internet creates new and unforeseen possibilities for moving scientific research in this direction. We no longer have to have a middleman, i.e. an intermediate agent between an individual scientist and the rest of the community or the public (Voinov and Costanza, 1999). The traditional way of getting the scientific message out is to publish in journals, present at conferences, or write a book. Now anyone can publish on the web and sooner or later search engines will start picking up these findings and guiding the public towards them if they are of general interest. Of course, there are pitfalls in this trend because it can result in propagation of misinformation and bad science, but there is also tremendous benefit that can be derived from rapid dissemination and

a much larger diversity of information sources. This system is parallel to traditional scientific peer review and may be considered complementary in many respects.

Even within the more traditional modes of scientific information sharing there have been some exciting developments. Peer-review journals are moving rapidly toward on-line publication and some have even adopted an open, online peer-review process (see, for example, journal *Biogeosciences* at <http://www.copernicus.org/EGU/bg/bg.html>). Scientists have also started sharing papers like people share music, i.e. by freely exchanging electronic reprints over the web. By analogy, perhaps a torrent/P2P application could be used to find and disseminate publications over the web. All researchers already have a collection of files on their computers that contain their own publications and perhaps papers that they have found interesting and downloaded from somewhere else. Scientists could share these libraries, rendering expensive journals obsolete. We already see a number of open access (free) scientific journals on the web, such as First Monday (<http://www.firstmonday.org/>), the Living Reviews series (<http://www.livingreviews.org>), Scientia Marina (<http://www.icm.csic.es/scimar>) and Ecology and Society (<http://www.ecologyandsociety.org>). This is an exciting trend that is likely to grow as we move to fully electronic publications. Hopefully publishing houses will be more flexible than the Recording Industry Association of America and the Motion Picture Association of America, and will adapt to this new environment without waging wars and lawsuits against researchers and software developers. Or maybe not: there are several lawsuits against Google already brewing to oppose their effort of scanning books and making book content available over the Internet (von Bubnoff, 2005).

We already witness how research communities are organised spontaneously around certain topics, and how group initiatives similar to research projects are developed. Consider, for instance, the Oil Drum project that currently is developed at <http://www.theoil Drum.com/>. This is a self-organised group of people who share similar views and concerns that are working on various issues that interest them. They publish data and findings on their blog for anyone to see and participate. There is an active community that is engaged in discussions, and that posts comments and questions, which further enhance and direct the research. All this is done on a totally volunteer basis. Another example is the on-line research spearheaded by Dr. Henry Niman, who analyses the dynamics of bird-flu with a blog of his own, where volunteers can help track local press and radio reports to understand the trends of the epidemic (Recombinomics: <http://www.recombinomics.com/>). Originally ridiculed by WHO (Zamiska, 2006) the results of this analysis have been gradually validated by more traditional studies of bird-flu. Some of the predictions of Niman have been reported to be even more accurate than the official science (McNeil, 2006).

There is an ongoing discussion about “blogs” vs. “journalism,” where journalists are rightfully concerned that the uncontrolled and unreviewed flow of news may very easily degenerate to the level of reporting rumours, i.e. without a qualified gatekeeper the quality of information will degrade. Similar issues are haunting scientists, who think that blogs may be polluting the information field with untested and useless “scientific” information. However, this opinion is countered by scien-

tists who frequent the 'blogosphere' and see that the dynamic hierarchy of links and recommendations generated by blogs creates a powerful collaborative filtering process that can surpass the traditional peer review process. And the more bloggers there are in a particular community, the more efficient this filtering becomes, actually reducing information overload (Butler, 2005). Could blogs be harbingers of a new model of scientific communication in the future where open-source research is reviewed and distributed over the Internet? Unfortunately, standard methods of accounting for scientific success do not account for participation in this kind of research. However, in terms of impact and importance, we would argue that this kind of activity deserves as much recognition as the highly desired publications in recognised peer-review journals. These standards will need to change.

We see the future of science moving strongly toward more collaborative and open research where data, code and credit are much more widely shared, and that embraces alternative modes of self-organised and community driven research. In this new scientific era the number of hits on individual home pages, and numbers of posts on scientific blogs, will become just as important indicators of scientific success as the numbers of publications in "Science" or "Nature." "In the new world-view, the universe is seen as a dynamic web of interrelated events. None of the properties of any part of this web is fundamental; they all follow from the properties of the other parts and the overall consistency of their mutual interrelations determines the structure of the entire web" (Capra, 1975). Clearly, we are entering an era, where the free flow of information will be crucial in order to tackle the pressing global problems we face. The complexity of the problems and associated models and data sets will require well-coordinated team efforts where individual scientists are best recognised and valued for their ability to contribute to the team and share their models, data and ideas.

REFERENCES

- Bollier, D., 1999. The power of openness. Why citizens, education, government and business should care about the coming revolution in open source code software. <http://h2oproject.law.harvard.edu/opencode/h2o/#25>.
- Butler, D., 2005. Joint efforts. *Nature* 438, 548–549.
- Capra, F., 1975. *The Tao of Physics*. Shambhala, 366 pp.
- Daly, H., Farley, J., 2003. *Ecological Economics*. Island Press.
- Drahoš, P., Braithwaite, J., 2002. *Information Feudalism: Who Owns the Knowledge Economy?* The New Press, New York.
- Hippel, E., Krogh, G., 2003. Open source software and the "Private–Collective" innovation model: Issues for organization science. *Organization Science* 14 (2), 209–223.
- Howard, J., 2005. *The Emerging Business of Knowledge Transfer: Creating Value from Intellectual Products and Services*. Australian Government Department of Education, Science and Training, Canberra.
- Kipp, M.E.I., 2005. Software and seeds: Open source methods. *First Monday* 10 (9). http://firstmonday.org/issues/issue10_9/kipp/index.html.
- Levy, S., 1984. *Hackers*. Anchor/Doubleday, New York.
- McNeil, D., 2006. Human flu transfers may exceed reports. *New York Times*, June 4, 2006.

- Markoff, J., 2006. Software out there. *The New York Times*, April 5, 2006. <http://www.nytimes.com/2006/04/05/technology/techspecial4/05lego.html?ex=1146628800&en=d6728f2af081fb16&ei=5070>.
- National Academy of Engineering, 2003. *The Impact of Academic Research on Industrial Performance*. National Academies Press, Washington.
- Nature, 2005. Let data speak to data. *Nature* 438 (7068), 531, doi:10.1038/438531a.
- Newscom, 2005. The open-source patent conundrum. http://news.com.com/2102-1071_3-5557340.html?tag=st.util.print.
- Newsforge, 2004. HP memo forecasts MS patent attacks on free software. <http://www.newsforge.com/article.pl?sid=04/07/19/2315200>.
- Owen-Smith, J., Powell, W.W., 2000. To patent or not to patent: Faculty decisions and institutional success at technology transfer. *Journal of Technology Transfer* 26, 99–114.
- Perens, B., 1998. The open source definition. <http://perens.com/Articles/OSD.html>.
- Perens, B., 2006a. Software patents vs. free software. <http://perens.com/Articles/Patents.html>.
- Perens, B., 2006b. The problem of software patents in standards. <http://perens.com/Articles/PatentFarming.html#2>.
- Raymond, E., 2000a. A brief history of hackerdom. <http://www.catb.org/~esr/writings/cathedral-bazaar/hacker-history/index.html>.
- Raymond, E., 2000b. Homesteading the Noosphere. <http://www.catb.org/~esr/writings/cathedral-bazaar/homesteading/>.
- Schweik, C., Evans, T., Grove, J., 2005. Open source and open content: A framework for global collaboration in social-ecological research. *Ecology and Society* 10 (1), 33 [online]. URL: <http://www.ecologyandsociety.org/vol10/iss1/art33/>.
- Tuomi, I., 2004. Knowledge sharing and the idea of the public domain. UNESCO 21st Century Dialogues. Building World Knowledge Societies, Joint Research Centre, Institute for Prospective Technological Studies, Seoul, Korea.
- Voinov, A., 2002. Teaching and learning ecological modelling over the web: A collaborative approach. *Conservation Ecology* 6 (1), 10 [online]. URL: <http://www.consecol.org/vol6/iss1/art10>.
- Voinov, A., Costanza, R., 1999. Watershed management and the web. *Journal of Environmental Management* 56, 231–245.
- von Bubnoff, A., 2005. The real death of print. *Nature* 438, 550–552.
- Wheeler, D., 2005. Why open source software/free software (OSS/FS, FLOSS, or FOSS)? Look at the numbers! <http://www.dwheeler.com>.
- Zamiska, N., 2006. Blogging biochemist tracks bird flu, but scientists remain skeptical. *The Wall Street Journal*, March 24–26, 2006, p. 28. http://agonist.org/20060323/niman_a_bird_flu_watcher_develops_a_following_through_the_internet.